Alissa McBain

Wrangle and Analyze Data – WeRateDogs

WeRateDogs is a hilarious Twitter account, and I'm glad Udacity chose to use it for this data wrangling project. I completed my project first by gathering the data from three sources, then assessing the data to see what needed to be cleaned up, and finally executing those cleaning steps.

First, I had to gather the data. There were three pieces of data to gather, and one of them proved a tough nut to crack. The first dataset, called WeRateDogs Twitter Archive ('wrd'), and the second dataset, Image Predictions, were fairly straight-forward to pull into my workspace. But, the third dataset, Additional Twitter data ('add_tweet_data'), needed to be pulled from the Twitter API, which was a new process to me. But, with the help of my mentor, and lots of Googling, I was able to pull it in and begin the next step of assessing.

To asses each dataset I used a variety of Python commands, most of them in the Pandas library. `df.info()` and `df.head()` provided the most useful data, but I also found `df.sample(n)`, `df['columnname'].value_counts()` and `df[df.columnname.duplicated()]` incredibly useful.

Cleaning the data was the most time consuming of the three steps. I cleaned eight quality issues and two tidiness issues. The quality issues ranged from removing retweets (as per the project instructions), changing datatypes, adjusting ratings (in the case of multiple dog photos) and removing duplicates. The tidiness issues I cleaned up were unstacking several variables that would be better represented as one variable, and combining all three datasets into one.

The most challenging issue to clean up for me was the actual ratings themselves. I found a pattern where multiple dog photos had ratings that were multiples of a valid rating. For example, a photo that may have been rated 13/10, but had four dogs in it was instead rated 52/40. I had to isolate these multiple dog photos, divide their denominators by ten to determine what the multiplying factor was, and then divide the numerator by that factor. Then, overwrite the incorrect rating with this new rating.

Along those same lines, I found a few entries where a non-rating had been picked up by the previously used algorithm that was designed to pick up ratings. I assume this algorithm was looking for a string of "n/n" (where n represents some number), because it picked up a couple of dates (ie, 3/15 for March 15th). A few of these incorrect ratings didn't actually have ratings in the tweet, therefore didn't apply

to our dataset, and needed to be dropped. And a few of these tweets had ratings, but they were later in the text string, so I had to figure out a way to find the ***last*** occurrence of "n/n" and replace the ratings with those numbers.

All in all, it was good challenge to gather, asses and clean these three data sets, I'm glad I was able to get through it, and now I have the confidence to be able to do this in a professional setting.